

HOW TO EXPLAIN SELENIUM AUTOMATION FRAMEWORK

1. **Programming Language**: Java.
2. **Framework Type**: Data Driven Framework, keyword driven by using Page Object Model.
3. **POM**: As per the Page Object Model, we have maintained a class for every web page. Each web page has a separate class and that class holds the functionality and members of that web page. Separate classes for every individual test.
4. **Packages**: We have separate packages for pages and tests. All the web page related classes come under the pages package and all the tests related classes comes under tests package. For example, Payment page and Login page have separate classes to store element locators.
5. **Properties File**: This file (config. properties) stores the information that remains static throughout the framework such as browser-specific information, application URL, development, testing URL, Screenshots path, security questions and answers, username, password etc.
6. We kept all test classes under src/test/java and remaining files like config.properties, POM classes, test data we kept under src/main/java.
7. **TestNG** – We are using testNG for assertions, grouping and parallel execution purpose.
8. **Screenshots** – Screenshots will be captured and stored in a separate folder and also the screenshots of failed test cases will be added to extent reports.
9. **Maven** – We are using Maven for build, execution and dependency purpose. WE integrated the TestNG dependency in POM.xml file and running this POM.xml file using Jenkins.
10. **Version Control Tool** – We use Git as a central repository to store our test scripts.
11. **Jenkins** – By using Jenkins CI tool, we execute test scripts on a daily basis and also for nightly execution based on the schedule. Test results will be sent to the peers using Jenkins.

12. Src/test/resources – We store AutoIT scripts, Sikuli images, license to make your application under test up and running files kept.

13. Test Base Class – Test Base Class (TestBase.java) deals with all the common functions used by all the pages. This class is responsible for loading the configurations from properties file, initializing the WebDriver, implicit waits, Extent Reports.